



2020 Fall - ECE 2220 Laboratory 1

Introduction to Quartus II and Programming the Lab Hardware

In this lab, using the Quartus II tutorial as a guide, you will learn the basics of using the Quartus® II software to create (using Verilog), simulate and implement a few very simple logic circuits on the DE10 standard board. Read and review the *Quartus Prime Introduction*.

- a) Create a new project and in a separate folder. The easiest way to do this is using the “New Project Wizard”. Save only one project in each folder. Choose “5CSXFC6D6F31C6” (about 6th from the bottom of the list) as the device in “Assign” menu.

NOTE: The name of the Verilog file must be the same as the name used for the top module.

- b) Create a new Verilog file (File-New-Verilog HDL). A new “Verilog” window will open. Implement the following in Verilog
 - i. Create two input signals “x” and “y”
- c) Using the structural formation, implement x **AND** y, x **OR** y, and x **XOR** y. Create (and show) a “Simulation Waveform Editor” File (New-University Program VWF) and a waveform window will open.
 - i. Create the waveforms for the inputs, x and y and run a “functional” simulation for the 3 logic functions.
- d) Use the switches SW[0] and SW[1] as the inputs and the LEDR[0], LEDR[1], and LEDR[2] for the output of the three logic functions; for x **AND** y, x **OR** y, and x **XOR** y. The pin assignments may be found either in the DE10 manual or in the DE10-Standard Pin Assignments file. Compile the code and download it to the DE10 board. (You will need to follow the directions in Section 3.3 of the DE10 manual closely particularly the part to “Configure the FPGA in JTAG Mode”).

***Note:** You will notice that the other LEDS seem to be on slightly. This is due to the fact that the default output is for a “weak” pull-up. The other LEDS can be turned off: Go back to the top level design. Click the following; 1) “Assignments”, 2) “Device”, 3) “Pin and Device Options” and then 4) “Unused Pins”. Change this setting to “As input tri-stated”.

- e) The following circuit is for a 1 bit full adder. Partial Verilog code for it is written in structural formation. The **wire** element used in the highlighted section of Verilog code is a keyword used to connect input and output ports of a module instantiation together with some other element in your design. Complete the code using similar logic as in the partially written code.

Use the switches SW[0], SW[1] and SW[2] as the inputs and the LEDR[0] and LEDR[1] for the outputs of sum and cout. Compile the code and download it to the DE10 board.

```
module full_adder ( a, b, cin, sum , cout );
```

```
    input a, b, cin;
```

```
    output sum, cout;
```

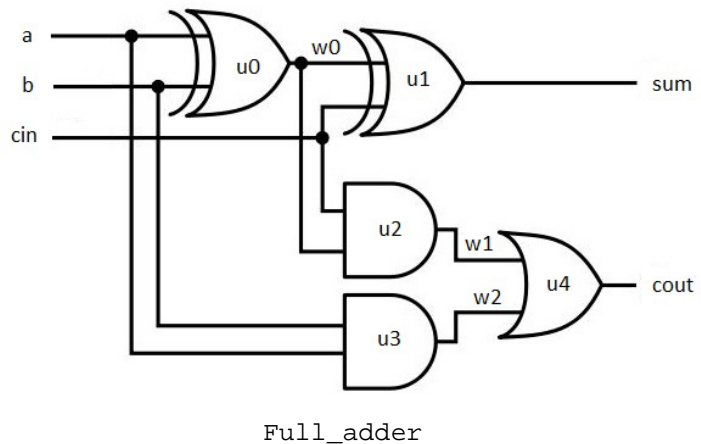
```
    wire w0,w1,w2;
```

```
    xor u0(w0,a,b);
```

```
    xor u1(sum,w0,cin);
```

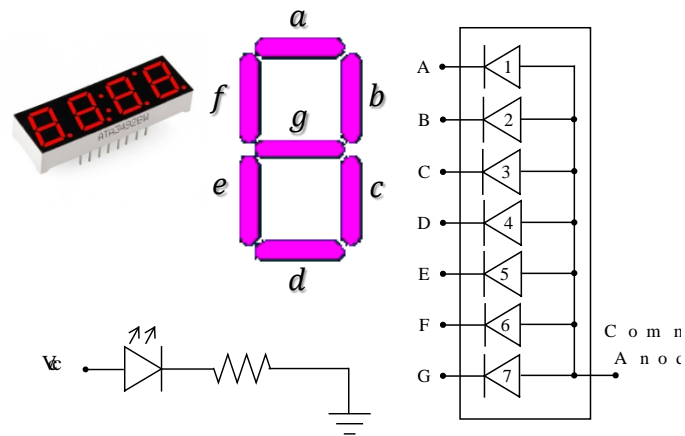
```
    ....
    ....
    ....
    ....
```

```
endmodule
```



Seven-segment displays (SSD) are commonly found on computers, watches, VCRs, monitoring devices, and other electronic devices to display numbers and some characters. The seven-segment displays used in the lab consist of seven independent Light Emitting Diodes (LEDs) in the configuration of a number "8". The different segments can be illuminated to display different numbers and letters. The segments of a seven-segment display are illustrated in the figure below. The SSDs, in general, come in packages with either a common anode or a common cathode. The SSDs on the DE10 board are common-anode. In this format the LED turns on with negative logic – i.e. low.

- f) Using an **always** block, and **if**, **else if** and **else** statements place the output of your full adder on to the right-most 7-segment display in a decimal format. Remember procedural statement (like **if**) must be inside an **always** block and if there are more than one statements it must contain **begin** and **end** statements as well. The 7-segment is driven by negative logic. To turn a segment on it must be give a “0”. Compile and download your code to the DE10 board.



Common-anode SSD display